IBM® Netcool® Netcool/OMNIbus Probe for
Genband IEMS
1.0

Reference Guide
May 25, 2012

IBM

Note

Before using this information and the product it supports, read the information in Appendix A, "Notices and Trademarks," on page 19.

Contents

About this guide

The following sections contain important information about using this guide.

Document control page

Use this information to track changes between versions of this guide.

The IBM Tivoli Netcool/OMNIbus Probe for Genband IEMS documentation is provided in softcopy format only. To obtain the most recent version, visit the IBM® Tivoli® Information Center:

http://publib.boulder.ibm.com/infocenter/tivihelp/v8r1/index.jsp?topic=/com.ibm.tivoli.namomnibus.doc/welcome_ptsm.htm

Table 1. Document modification history		
Document version	**Publication date**	**Comments**
SC27-4375-00	May 25, 2012	First IBM publication.

Conventions used in this guide

All probe guides use standard conventions for operating system-dependent environment variables and directory paths.

Operating system-dependent variables and paths

All probe guides use standard conventions for specifying environment variables and describing directory paths, depending on what operating systems the probe is supported on.

For probes supported on UNIX and Linux operating systems, probe guides use the standard UNIX conventions such as **$***variable* for environment variables and forward slashes (**/**) in directory paths. For example:

`$OMNIHOME/probes`

For probes supported only on Windows operating systems, probe guides use the standard Windows conventions such as **%***variable***%** for environment variables and backward slashes (****) in directory paths. For example:

`%OMNIHOME%\probes`

For probes supported on UNIX, Linux, and Windows operating systems, probe guides use the standard UNIX conventions for specifying environment variables and describing directory paths. When using the Windows command line with these probes, replace the UNIX conventions used in the guide with Windows conventions. If you are using the bash shell on a Windows system, you can use the UNIX conventions.

Note : The names of environment variables are not always the same in Windows and UNIX environments. For example, %TEMP% in Windows environments is equivalent to $TMPDIR in UNIX and Linux environments. Where such variables are described in the guide, both the UNIX and Windows conventions will be used.

Operating system-specific directory names

Where Tivoli Netcool/OMNIbus files are identified as located within an *arch* directory under NCHOME or OMNIHOME, *arch* is a variable that represents your operating system directory. For example:

`$OMNIHOME/probes/`*arch*

The following table lists the directory names used for each operating system.

Note : This probe may not support all of the operating systems specified in the table.

Table 2. Directory names for the arch variable	
Operating system	**Directory name represented by** *arch*
AIX® systems	`aix5`
Red Hat Linux® and SUSE systems	`linux2x86`
Linux for System z	`linux2s390`
Solaris systems	`solaris2`
Windows systems	`win32`

OMNIHOME location

Probes and older versions of Tivoli Netcool/OMNIbus use the OMNIHOME environment variable in many configuration files. Set the value of OMNIHOME as follows:

- On UNIX and Linux, set $OMNIHOME to $NCHOME/omnibus.
- On Windows, set %OMNIHOME% to %NCHOME%\omnibus.

Chapter 1. Probe for Genband IEMS

The Probe for Genband IEMS is software solution for voice over packet networks providing network management functions. The Probe for Genband IEMS monitors SNMP traps and events generated by a Genband IEMS server on both UDP and TCP sockets concurrently.

This guide contains the following sections:

Summary

Each probe works in a different way to acquire event data from its source, and therefore has specific features, default values, and changeable properties. Use this summary information to learn about this probe.

The following table provides a summary of the Probe for Genband IEMS.

Table 3. Summary	
Probe target	Genband IEMS CVM14 and CVM15
Probe executable name	`nco-p-genband-iems`
Probe installation package	`omnibus-`*`arch`*`-probe-nco-p-genband-iems-`*`version`* **Note :** To successfully install the Probe for Genband IEMS you must uninstall the Probe for Nortel IEMS if it exists on the operating system. If the Probe for Nortel IEMS is installed on the system it can result in the probe binaries not installing in the following default location and these errors may not be reported in the installation log: - `$OMNIHOME/probes/` - `$OMNIHOME/probes/`*`arch`*`/`
Package Version	1.0
Probe supported on	For details of supported operating systems, see the following Release Notice on the IBM Software Support website: https://www-304.ibm.com/support/docview.wss?uid=swg21591661
Properties file	`$OMNIHOME/probes/`*`arch`*`/genband_iems.props`
Rules file	`$OMNIHOME/probes/`*`arch`*`/genband_iems.rules`

Table 3. Summary (continued)	
Requirements	A currently supported version of IBM Tivoli Netcool/OMNIbus `common-libcrypt-1_0` is required for Netcool/OMNIbus version 7.1 and version 7.2 (but not for V7.2.1).
Connection method	TCP/IP, UDP
Multicultural support	Not Available
Peer-to-peer failover functionality	Available
IP environment	IPv4 and IPv6 **Note :** The probe is supported on IPv6 when running on IBM Tivoli Netcool/OMNIbus V7.3.0, 7.3.1 and 7.4.0.
Federal Information Processing Standards (FIPS)	IBM Tivoli Netcool/OMNIbus uses the FIPS 140-2 approved cryptographic provider: IBM Crypto for C (ICC) certificate 384 for cryptography. This certificate is listed on the NIST website at http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/1401val2004.htm. For details about configuring Netcool/OMNIbus for FIPS 140-2 mode, see the *IBM Tivoli Netcool/OMNIbus Installation and Deployment Guide*.

Installing probes

All probes are installed in a similar way. The process involves downloading the appropriate installation package for your operating system, installing the appropriate files for the version of Netcool/OMNIbus that you are running, and configuring the probe to suit your environment.

The installation process consists of the following steps:

1. Downloading the installation package for the probe from the Passport Advantage Online website.

   Each probe has a single installation package for each operating system supported. For details about how to locate and download the installation package for your operating system, visit the following page on the IBM Tivoli Knowledge Center:

   http://www-01.ibm.com/support/knowledgecenter/SSSHTQ/omnibus/probes/all_probes/wip/reference/install_download_intro.html

2. Installing the probe using the installation package.

   The installation package contains the appropriate files for all supported versions of Netcool/OMNIbus. For details about how to install the probe to run with your version of Netcool/OMNIbus, visit the following page on the IBM Tivoli Knowledge Center:

   http://www-01.ibm.com/support/knowledgecenter/SSSHTQ/omnibus/probes/all_probes/wip/reference/install_install_intro.html

3. Configuring the probe.

   This guide contains details of the essential configuration required to run this probe. It combines topics that are common to all probes and topics that are peculiar to this probe. For details about additional configuration that is common to all probes, see the *IBM Tivoli Netcool/OMNIbus Probe and Gateway Guide*.

Configuration and tuning

After you have installed the Probe for Genband IEMS, you might need to make some configuration changes to suit your environment.

If you are running the probe on a system where another trapd process is running (for example, HP NNM, or SunNet Manager), you must ensure that a different SNMP port is specified on the command line or by defining the **Port** property.

Note : You require two licenses if you set the Probe for Genband IEMS to accept both UDP and TCP traps.

Performance tuning

The following guidelines can help you to optimize performance:

Buffer settings
To increase the efficiency of sending alerts to the ObjectServer, the following properties are available:

- **Buffering** - When set to 1, this property instructs the probe to send alerts when the internal alert buffer has reached the size specified by the **BufferSize** property.
- **BufferSize** - This property specifies the size of the buffer that the probe uses to store alerts before sending them to the ObjectServer.
- **FlushBufferInterval** - This property specifies an interval in seconds that the probe waits before flushing the alerts to the ObjectServer. It limits the time that alerts wait in the buffer when the buffer has yet to reach the size specified by the **BufferSize** property.

Note : The internal alert buffer contains alerts that have been derived from the raw SNMP traps.

Trap queue size setting
The probe maintains a queue that is used to store raw SNMP traps before they are processed by the probe. When an event storm occurs, this queue can grow quickly consuming excessive amounts of memory. To prevent this, you can use the **TrapQueueMax** property to specify a maximum size to which this queue can grow before the probe starts to discard traps.

Note : The **TrapQueueMax** property is set to 20000 by default. If the value is set to 0, the probe sends the following warning message: `Memory growth of the probe is unbounded.`

Message level and raw capture settings
When using the probe in production mode, do not use the following settings:

- `MessageLevel: debug`
- `RawCapture: 1`

IP Address resolution
If the Domain Name Server (DNS) is not resolving IP addresses quickly, you can improve performance by setting the **NoNameResolution** property to 0; this uses less memory.

Rules file
Avoid using construct details (`$*`) in the rules file.

Alert conversion
To configure the Probe for Genband IEMS to convert alerts to Genband IEMS use the following example:

- `$OMNIHOME/bin/nco_sql-user` *username* `-server` *servername*
- `UPDATE alert.conversions SET Conversion = '`*Genband IEMS*`' where Value =` *5155*
- `go`

Data acquisition

Each probe uses a different method to acquire data. Which method the probe uses depends on the target system from which it receives data.

The Probe for Genband IEMS is a direct SNMP monitoring probe. The probe acquires event data by acting as a trap daemon and monitoring SNMP traps and events on both UDP and TCP sockets. It is able to handle a high volume and high rate of traps and receives traps independently of trap processing using an internal queue mechanism.

Note : Depending on the Genband IEMS version the following elements may not be available in the EMS. Check with Genband for further details:

- nnExtAlarmActiveAdditionalText2
- nnExtAlarmActiveIdAlias

The probe handles high trap rates and high burst rates using two buffers: one buffer for all of the sockets that the probe monitors, and an internal queue between the reader and writer sides of the probe.

Data acquisition is described in the following topics:

- "Resynchronization" on page 4
- "Peer-to-peer failover functionality" on page 4

Resynchronization

When the probe starts, it performs a full resynchronization with the Genband IEMS server to collect all current alarms. If the full resynchronization fails, the **RedoResynch** property allows you to specify whether the probe reattempts a full resynchronization. Additionally, the **ResynchAttempt** property allows you to specify the number of full resynchronization attempts that the probe makes.

Note : You can instruct the probe to not perform a full resynchronization on startup by setting the **DisableResynch** property to 1.

Following resynchronization, the probe processes each alarm as it is generated. The Genband IEMS server maintains a cache table of the most recent alarms. The number of alarms stored in the cache table is a configurable value on the Genband IEMS server. To ensure no alarms are lost during resynchronization specify the **NlmConfigGlobalEntryLimit** property for the Probe for Genband IEMS to the same value.

As alarms are generated, they are each allocated a sequence number. This enables the probe to recognize when an alarm has been missed. If an alarm is missed, the probe performs a resynchronization. If the number of alarms missed is less than or equal to the number of alarms stored in the cache table, the resynchronization is incremental. If the number of missed alarms exceeds the number of alarms stored in the cache table, a full resynchronization is performed.

Peer-to-peer failover functionality

The probe supports failover configurations where two probes run simultaneously. One probe acts as the master probe, sending events to the ObjectServer; the other acts as the slave probe on standby. If the master probe fails, the slave probe activates.

While the slave probe receives heartbeats from the master probe, it does not forward events to the ObjectServer. If the master probe shuts down, the slave probe stops receiving heartbeats from the master and any events it receives thereafter are forwarded to the ObjectServer on behalf of the master probe. When the master probe is running again, the slave probe continues to receive events, but no longer sends them to the ObjectServer.

Example property file settings for peer-to-peer failover

You set the peer-to-peer failover mode in the properties files of the master and slave probes. The settings differ for a master probe and slave probe.

Note : In the examples, make sure to use the full path for the property value. In other words replace $OMNIHOME with the full path. For example: /opt/IBM/tivoli/netcool.

The following example shows the peer-to-peer settings from the properties file of a master probe:

```
Server      :      "NCOMS"
RulesFile   :      "master_rules_file"
MessageLog  :      "master_log_file"
PeerHost    :      "slave_hostname"
PeerPort    :      6789 # [communication port between master and slave probe]
Mode        :      "master"
PidFile     : "master_pid_file"
```

The following example shows the peer-to-peer settings from the properties file of the corresponding slave probe:

```
Server      :      "NCOMS"
RulesFile   :      "slave_rules_file"
MessageLog  :      "slave_log_file"
PeerHost    :      "master_hostname"
PeerPort    :      6789 # [communication port between master and slave probe]
Mode        :      "slave"
PidFile     : "slave_pid_file"
```

Properties and command line options

You use properties to specify how the probe interacts with the device. You can override the default values by using the properties file or the command line options.

The following table describes the properties and command line options specific to this probe. For information about default properties and command line options, see the *IBM Tivoli Netcool/OMNIbus Probe and Gateway Guide,* (SC14-7530).

Table 4. Properties and command line options		
Property name	**Command line option**	**Description**
AlarmIndexIsOID *boolean*	-alarmindexlsoid *boolean*	Use this property to specify whether the probe treats the resource identifier in the resynchronization table as an object identifier (OID). 0: The probe does not treat the resource identifier in the resynchronization table as an OID. 1: The probe treats the resource identifier in the resynchronization table as an OID. The default is 0. **Note :** This property is only used when the **GenbandCVMVersion** property is set to a value lower than 14. For **GenbandCVMVersion** versions 14 and above, the probe always treats alarm resource IDs as an OID.
BindAddress *string*	-bindaddress *string*	Use this property to specify the IP address to which the socket is bound. The default is " ".

Property name	Command line option	Description
ConfPath *string*	`-confpath` *string*	Use this property to specify a colon separated list of the paths that contain configuration files for the SNMP probe engine. The default is `'$OMNIHOME/probes/`*arch*`:/export/build/tmp/$`{*USERNAME*}`'`
DisableResynch *integer*	`-disableresynch` *integer*	By default the probe performs a full resynchronization at startup, use this property to disable this automatic resynchronization. This property takes the following values: 0: The probe preforms a full resynchronization at startup. 1: The probe does not perform synchronization on startup. The default is 0.
FlushBufferInterval *integer*	`-flushbufferinterval` *integer*	Use this property to specify how often (in seconds) the probe flushes all alerts in the buffer to the ObjectServer. The default is 0 (which instructs the probe to never flush the alerts to the ObjectServer).
GenbandCVMVersion *integer*	`-genbandcvmversion` *integer*	Use this property to specify the version of Genband CVM to which the probe is connected. The default is 14.
GenbandEMSCommunity *string*	`-genbandemscommunity` *string*	Use this property to specify the SNMP community string for Genband IEMS. The default is `'public'`.
GenbandEMSHostName *integer*	`-genbandemshostname` *integer*	Use this property to specify the IP address of the host running Genband IEMS to which the probe connects. The default is 127.0.0.1.
GenbandEMSPort *integer*	`-genbandemsport` *integer*	Use this property to specify the port number of the Genband IEMS host to which the probe connects. The default is 8001.

Table 4. Properties and command line options (continued)

Property name	Command line option	Description
Heartbeat *integer*	`-heartbeat` *integer*	Use this property to specify the frequency (in seconds) that the probe ends a heartbeat message to the Genband IEMS server to check if Genband IEMS is still running. The default is 60.
MIBDirs *string*	`-mibdirs` *string*	Use this property to specify a colon-separated list of directories in which the probe searches for MIB modules. The default is `'$OMNIHOME/common/mibs'`.
MIBFiles *string*	`-mibfiles` *string*	Use this property to specify the name of the MIB file. The default is `'$OMNIHOME/probes/mib.txt'`. **Note :** If you are using a rules file generated by the trapd converter, you must set this property to point to an empty file; for example, `/dev/null`.
MIBs *string*	`-mibs` *string*	Use this property to specify a colon-separated list of the MIB modules that the probe loads. The default is `'ALL'` (which instructs the probe to load all modules available in the directories specified by the **MIBDirs** property).
NlmConfigGlobal EntryLimit *integer*	`-nlmconfigglobal entrylimit` *integer*	Use this property to specify the maximum number of alarms that Genband IEMS stores in its cache of most recently generated alarms. This is effectively the number of consecutive missed events that the probe allows before it performs a full resynchronization. Otherwise, the probe performs an incremental resynchronization. The default is 25.

Table 4. Properties and command line options (continued)

Table 4. Properties and command line options (continued)		
Property name	**Command line option**	**Description**
NoNameResolution *boolean*	-nonameresolution *boolean*	Use this property to specify whether the probe performs name resolution on IP addresses. 0: The probe performs name resolution. 1: The probe does not perform name resolution. The default is 0.
NonPrintableAsHex *boolean*	-nonprintableashex *boolean*	Use this property to specify whether the probe sets non-printable characters to their hexadecimal values. 0: The probe does not set non-printable characters to their hexadecimal values. 1: The probe sets non-printable characters to their hexadecimal values. The default is 0. **Note :** This property only effects varbinds of octet string type. If this property is set to 1 it will assign event tokens with the hexadecimal value of non-printable characters, but the probe does not have this functionality.
PersistentDir *string*	-persistentdir *string*	Use this property to specify where the persistent configuration information is stored. The default is '$OMNIHOME/var'.
Port *integer*	-port *integer*	Use this property to specify the port to which the probe listens for SNMP traffic. The default is 162.

Property name	Command line option	Description
Protocol *string*	-protocol *string* • -udp (This is equivalent to **Protocol** with a value of UDP or -protocol UDP) • -tcp (This is equivalent to **Protocol** with a value of TCP or -protocol TCP) • -all (This is equivalent to **Protocol** with a value of ALL or -protocol ALL) • -any (This is equivalent to **Protocol** with a value of ANY or -protocol ANY) • -udp6 (This is equivalent to **Protocol** with a value of UDPIPv6 or -protocol UDPIPv6) • -allipv6 (This is equivalent to **Protocol** with a value of TCPIPv6 and UDPIPV6 or -protocol ALLIPv6)	Use this property to specify the network protocol that the probe uses. If the probe is running in an IPv4 environment, specify one of the following values: • UDP • TCP • ALL • ANY The values ANY and ALL are interchangeable. If the probe is running in an IPv6 environment, specify one of the following values: • UDPV6 • TCPV6 • ALLIPV6 Currently, only the Solaris version of the probe supports IPv6. The default is "UDP"
QuietOutput *boolean*	-quietoutput *boolean*	Use this property to specify whether the probe outputs tokens that correspond to an OID with symbolic OID expansion. Possible values are: 0: The probe outputs tokens with symbolic OID expansion. 1: The probe outputs tokens without symbolic OID expansion. The default is 1. **Note :** When the **QuiteOutput** property is set to 0 alarm resynchronization will fail because the probe will be unable to read OIDs with expanded symbols.

Table 4. Properties and command line options (continued)

Table 4. Properties and command line options (continued)		
Property name	**Command line option**	**Description**
RedoResynch *boolean*	`-redoresynch` *boolean*	Use this property to specify whether the probe reattempts a full resynchronization if it previously failed. 0: The probe does not attempt to do a full resynchronization if it previously failed. 1: The probe attempts to do a full resynchronization if it previously failed. The default is 0.
ResynchAttempt *integer*	`-resynchattempt` *integer*	Use this property to specify the number of full resynchronization attempts that the probe makes. The default is 0.
SleepTime *integer*	`-sleeptime` *integer*	Use this property to specify the polling interval (in seconds) of the trap list. If there are no traps to be processed, the probe sleeps for this amount of time before polling the trap queue again. The default is 1.
SocketSize *integer*	`-socketsize` *integer*	Use this property to specify the size (in bytes) of the kernel buffer on the socket being used. This is set on a per-socket basis. A higher value increases the number of traps that the probe can handle. For UDP traps, this improves performance. The minimum value is 128. The default is 8192.
TrapQueueMax *integer*	`-trapqueuemax` *integer*	Use this property to specify the maximum number of traps that can be queued for processing at any one time. The probe discards any traps received while the buffer is full. The default is 20000.

Running the probe

The way in which you run the probe depends on whether you are running UNIX or Windows.

To start the probe on UNIX, use the following command:

```
$OMNIHOME/bin/nco_p_[probe_name]
```

To start the probe from a Windows command prompt, use the following command:

```
%OMNIHOME%\bin\nco_p_[probe_name]
```

Elements

The probe breaks event data down into tokens and parses them into elements. Elements are used to assign values to ObjectServer fields; the field values contain the event details in a form that the ObjectServer understands.

The elements generated by the Probe for Genband IEMS are described in the following topics:

- "Elements Generated from IEMS Traps" on page 11
- "Resynchronization elements" on page 12
- "Dynamic elements" on page 13

Elements Generated from IEMS Traps

IBM Tivoli Netcool/OMNIbus Probe for Genband IEMSgenerates a set of elements for traps it receives from IEMS.

The following table describes the elements that the IBM Tivoli Netcool/OMNIbus Probe for Genband IEMS generates from the traps it receives from IEMS.

Table 5. IEMS trap elements		
Element name	**SNMP Version**	**Element description**
`$community`	V1 and V2c	This element contains the SNMP community string.
`$enterprise`	V1	This element contains the SNMP enterprise string.
`$EventCount`	V1 and V2c	This element contains the number of traps processed during the current execution of the probe.
`$generic-trap`	V1	This element contains the SNMP generic trap integer value.
`$IPaddress`	V1 and V2c	This element contains the IP address (origin of the SNMP trap).
`$Node`	V1 and V2c	This element contains the node name (origin of the SNMP trap). IP address (if node name cannot be resolved).
`$notify`	V2c	This elements contains the V2c specific field.
`$PeerAddress`	V1 and V2c	This element contains the host name or IP address where the SNMP trap was received from.
`$PeerIPaddress`	V1 and V2c	This element contains the IP address where the SNMP trap was received from.

Table 5. IEMS trap elements (continued)		
Element name	**SNMP Version**	**Element description**
$Protocol	V1 and V2c	This element contains the protocol of the trap received. This can be either UDP or TCP.
$ReceivedPort	V1 and V2c	This element contains the port number where the SNMP trap was received from. This is determined by the **Port** property.
$ReceivedTime	V1 and V2c	This element contains the time that the SNMP packet was received from the network interface.
$ReqId	V1	This element contains the SNMP request ID.
$SNMP_Version	V1 and V2c	This element has the value 1 for SNMP V1 traps and the value 2 for SNMP V2c traps.
$specific-trap	V1	This element contains the SNMP specific trap integer value.
$Uptime	V1 and V2c	This element contains the SNMP uptime for traps expressed in the format 0:00:00.
$UpTime	V1 and V2c	This element contains the SNMP uptime for traps expressed as an integer.

Resynchronization elements

The probe generates a set of elements for resynchronization alerts.

The following table describes the resynchronization elements that the Probe for Genband IEMS generates.

Table 6. Resynchronization elements	
Element name	**Element description**
$sysUpTime	This element contains the system up-time of the server that generated the alarm.
$trapOID	This element contains the object identifier of the trap.
$alarmActiveResourceId	This element contains the resource identifier of the active alarm.
$alarmActiveDateAndTime	This element contains the date and time of the active alarm.

Table 6. Resynchronization elements (continued)	
Element name	**Element description**
`$alarmActiveDescription`	This element contains a description of the active alarm.
`$nnExtAlarmActiveEventType`	This element contains the event type of the active alarm.
`$nnExtAlarmActiveIdAlias`	This element contains the alias name of the active alarm.
`$nnExtAlarmActiveProbableCause`	This element contains the probable cause of the active alarm.
`$nnExtAlarmActiveAdditionalText`	This element contains text associated with the active alarm.
`$nnExtAlarmActiveAdditionalText2`	This element contains additional text associated with the active alarm.
`$nnExtAlarmActiveDocumentation Pointer`	This element contains a documentation pointer of the active alarm.
`$nnExtAlarmActiveResource Description`	This element contains the description of the resource associated with the active alarm.
`$nnExtAlarmActiveManualClear`	This element indicates whether the active alarm can be cleared manually.
`$nnExtAlarmActiveSequenceNumber`	This element contains the sequence number of the active alarm.

Dynamic elements

The dynamic elements that the probe generates are entirely dependent on the devices monitored.

The varbind variables that are generated by the SNMP trap are mapped to elements called $1,$2,$3, and so on. For each varbind, the object ID is placed in a corresponding element called $OID1, $OID2, $OID3, and so on up to the number of varbind elements.

Note : Versions of the Probe for Nortel IEMS prior toversion 3.5) had no leading dot (.) in the $OIDn elements, whereas the latest probe does include the leading dot; if you are upgrading from an old version of the probe, your rules files may need updating.

The probe can also generate the following elements from various representations of the varbind variables:

- `$n_raw` - raw string representation of the varbind variables (containing all control characters)
- `$n_text` - printable text representation of the varbind variables (with non-printable characters replaced with periods)
- `$n_hex` - hexadecimal representation of the varbind variables

Note : The `$n_raw, $n_text` and `$n_hex` elements are only available for SNMP variables of type `OCTET-STRING`.

Generic Trap Handling

The probe handles each generic trap type differently.

The following table describes the handling of each generic trap type.

Table 7. Generic traps	
Generic trap	**Handling**
Generic trap-type 0 - Cold Start	Summary field set to `Cold Start` `AlertGroup` field set to `Generic` `Severity` field set to 4
Generic trap-type 1 - Warm Start	Summary field set to `Warm Start` `AlertGroup` field set to `Generic` `Severity` field set to 4

Generic trap-type 2 - Link Down	Summary field set to `Link Down`
	`AlertKey` field set to the $1 varbind (`ifIndex`)
	`AlertGroup` field set to `Generic`
	`Severity` field set to 5
	`Identifier` field set to `Node name` plus `Agent` plus `generic-trap` plus specific trap plus `ifIndex`

Generic trap-type 3 - Link Up	Summary field set to `Link Up`
	`AlertKey` field set to the $1 varbind (`ifIndex`)
	`AlertGroup` field set to `Generic`
	`Severity` field set to 2
	`Identifier` field set to `Node name` plus `Agent` plus `generic-trap` plus specific trap plus `ifIndex`

Generic trap-type 4 - Authentication	By default, Authentication traps are not discarded
Generic trap-type 5 - EGP Neighbor Loss	Summary field set to `EGP Neighbor Loss` `AlertGroup` field set to `Generic` `Severity` field set to 3

Error messages

Error messages provide information about problems that occur while running the probe. You can use the information that they contain to resolve such problems.

The following table describes the error messages specific to this probe. For information about generic error messages, see the *IBM Tivoli Netcool/OMNIbus Probe and Gateway Guide* (SC14-7608).

Table 8. Error messages		
Error	**Description**	**Action**
`Could not init the buffer`	The probe could not initialize the buffer on startup.	Check the value specified for the **`NlmConfigGlobal EntryLimit`** property.
`Error either the value is empty or nrc is not SUCCESS`	Either the probe could not send a GET request to the IEMS server or it failed to extract the OID from the GET reply from the IEMS server.	Check that the IEMS server is running correctly and received the GET request.

`Error when parsing the oid %s`
 	
`Expected varbind from get!`
	The probe encountered a problem while parsing the event.	Check that the IEMS server is running correctly.

`ExtractGenericGenband Alarm: NSProbeEventCreate failed`
 	
`GenerateIncrResynch Alert: NSProbeEventCreate failed`
 	
`Not enough memory`
	The probe has received more alarms than it can store in the buffer file.	Check the value specified by the **`TrapQueueMax`** property.
		
		Make more memory available.

`Exception value returned when attempting to do a GETNEXT: errorcode`
 	
`Invalid response to SNMP GETNEXT command`
	The probe is attempting to get the active alarm index. The *errorcode* value will contain the system's error code.	Check that the IEMS server is running correctly.

`ExtractGenericGenband Alarm 0: Error in issuing get req: errorcode`
 	
`ExtractGenericGenband Alarm 1: Failure when attempting to do SNMP GET`
 	
`ExtractGenericGenband Alarm 2: Error in issuing get req: errorcode`
	The probe is attempting to get the varbinds of resynchronized alarms. The*errorcode* value will contain the system's error code.	Check that the IEMS server is running correctly.

Table 8. Error messages (continued)		
Error	**Description**	**Action**

`GenerateResynchAlert: Failure when attempting to do SNMP GET`
 	
`GenerateResynchAlert: Error in issuing get req: %s`
 	
`GenbandIncremental Resynch: Error in issuing get req: ErrorcodeA seq_num: ErrorcodeB`
	The probe is sending PDU messages to the IEMS system requesting a trap OID for an alarm resynchronization operation. The *ErrorcodeA* and *ErrorcodeB* values will contain system error codes.	Check that the IEMS server is running correctly.

`NSProbeEventSetElement (ReqId): errorcode`	The probe failed to set an event token value. The *errorcode* value will contain the system's error code.	Make more memory available.
`Unable to create SNMP pdu`	The probe failed to process the alarms.	Check that Genband IEMS is running correctly.
`Unable to open snmp session`	The probe failed to connect to the Genband IEMS server.	Check that Genband IEMS is running correctly. Check the values specified for the following properties: • **ConfPath** • **GenbandCVMVersion** • **Protocol**

`Unable to parse oid %s`
 	
`Unrecognized ASN1 type`
	The probe cannot parse the alarm; this is probably because the alarm is not in a format that the probe can understand.	Check that the Genband IEMS server is running correctly.
		
		Check the setting of the **GenbandCVMVersion** property matches the Genband IEMS version.
		
		Check the details in this file: `$OMNIHOME/probes/arch/ mttrapd.conf`.

`Unable to extract oid`	When the probe started, it could not retrieve the existing alarms.	Check that the Genband IEMS server is running currently.
`Unexpected trap oid stored in nlmLogVariableTable discarding %s`	The incoming data stream may be corrupt.	Check that the ObjectServer is running correctly.
`Warning: OID not increasing: %s >= %s\n`	The probe is not receiving alarms from the Genband IEMS server.	Check that the IEMS server is running correctly.

ProbeWatch messages

During normal operations, the probe generates ProbeWatch messages and sends them to the ObjectServer. These messages tell the ObjectServer how the probe is running.

The following table describes the raw ProbeWatch error messages that the probe generates. For information about generic ProbeWatch messages, see the *IBM Tivoli Netcool/OMNIbus Probe and Gateway Guide* (SC14-7608).

Table 9. ProbeWatch messages

ProbeWatch message	Description	Triggers/causes

`GENBAND IEMS resynch Failed`
 	
`GENBAND incremental resynch failed`
	The probe failed to resynchronize.	The probe cannot connect to the IEMS sever.

`Finished GENBAND IEMS resynch`	The probe successfully performed a resynchronization.	The probe has just performed a resynchronization.

Appendix A. Notices and Trademarks

This appendix contains the following sections:

- Notices
- Trademarks

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing 2-31 Roppongi 3-chome, Minato-ku
Tokyo 106-0032, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who want to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
Software Interoperability Coordinator, Department 49XA

3605 Highway 52 N
Rochester, MN 55901
U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this information and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement, or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

All IBM prices shown are IBM's suggested retail prices, are current and are subject to change without notice. Dealer prices may vary.

This information is for planning purposes only. The information herein is subject to change before the products described become available.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

Each copy or any portion of these sample programs or any derivative work, must include a copyright notice as follows:

© (your company name) (year). Portions of this code are derived from IBM Corp. Sample Programs. © Copyright IBM Corp. _enter the year or years_. All rights reserved.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Trademarks

IBM, the IBM logo, ibm.com, AIX, Tivoli, zSeries, and Netcool are trademarks of International Business Machines Corporation in the United States, other countries, or both.

Adobe, Acrobat, Portable Document Format (PDF), PostScript, and all Adobe-based trademarks are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, other countries, or both.

Intel, Intel Inside (logos), MMX, and Pentium are trademarks of Intel Corporation in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

IBM.